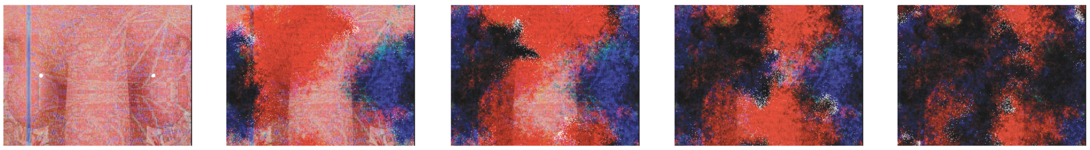
Introduction to Joseph Nechvatal's *Computer Virus Project 2.0*

by Joseph Nechvatal & Stephane Sikora

2001



Joseph Nechvatal, *Computer Virus Project 2.0 Attack* (2001)

Joseph Nechvatal's 2001 *Computer Virus Project 2.0* follows along the same lines as previous viral works created by Nechvatal in 1992 in Arbois, France: where an unpredictable progressive virus operated on a degradation-transformation of a Nechvatal art image.

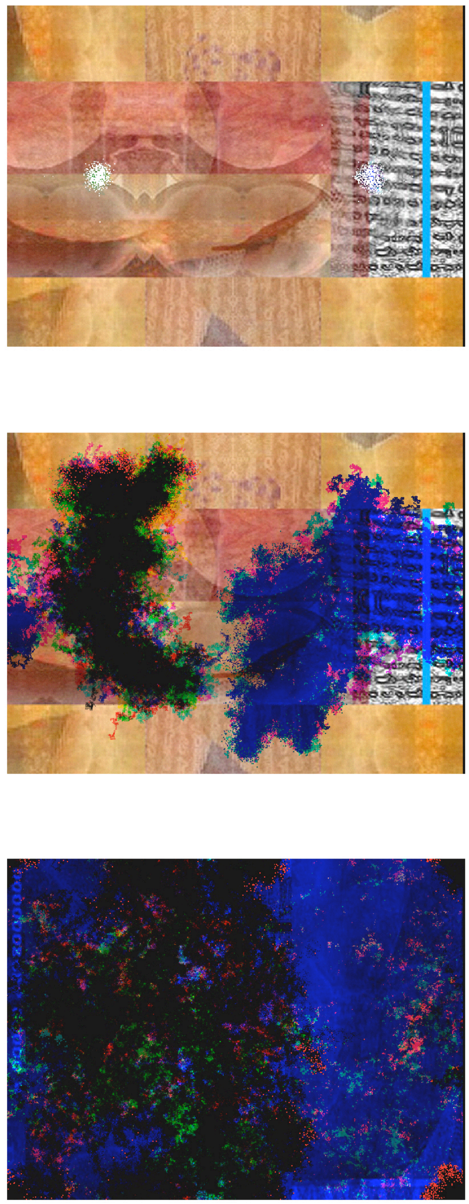
In 2001, using a C++ framework, Nechvatal and his programmer-collaborator Stephane Sikora have brought Nechvatal's *Computer Virus Project 1* (1992-1993) into the realm of artificial life (A-Life) (i.e. into a synthetic system that exhibits behaviors characteristic of natural living systems). With *Computer Virus Project 2.0*, elements of artificial life have been introduced in that viruses are modeled to be autonomous agents living in/off the image. The project simulates a population of active viruses functioning as an analogy of a viral biological system.

Computer Virus Project 2.0 actively propagates viral attacks on image-files from Nechvatal's recent *ec-satyricOn 2000 (enhanced)+ bodies in the bit-stream (compliant)* series. Here viral algorithms - based on a viral biological model - are used to define evolutionary processes which are then applied to image-files from Nechvatal's *ec-satyricOn 2000 (enhanced)+ bodies in the bit-stream (compliant)* series. Among the different techniques used here are models that result from embodied artificial intelligence and the paradigm of genetic programming.

The Model

The world is modeled as an image via a set of pixels. Every pixel's color is defined by R,G,B real number vectors which represent the red, green and blue components of every pixel's color.

The image world has no edges. Every square on the edge of the image is adjacent to another on the opposite edge. A virus can move around the image and impact the image world as different colors actually correspond to resources used for survival by the viruses.



Joseph Nechvatal, *Computer Virus Project 2.0 Attack II* (2001)

The Virus's Behavior

The behavior of a virus is modeled as a generated looping activity that is typical of situated artificial intelligence work. A virus will pick up information from its environment, decide on a course of action, and carry it out. The loop is simplified here because of the abstract character of the simulacrum. Viral instructions provide different possibilities for executing instructions according to the environmental conditions in which the virus is living.

A virus will perceive the pixel it is on and the eight adjacent ones. It can get information on its color and on the possible presence of other viruses. In order to decide on a course of action, each virus is programmed with a set of randomized instructions of different kinds; some relate to direction, others to a change in the color of the current pixel (the one the virus is in). Others control the implementation of the program and carry out tests.

Once the program has been executed, following actions to be carried out randomly arise. As the virus executes them, it moves to one of the adjacent squares and changes the current pixel. It can even reproduce itself (reproduction here results from the instruction 'divide'. A virus that carries out that instruction will produce a replica of itself - although slightly altered. Its genome-program changes with the mutation operator).

In addition to these changes, every cycle produces a change in the energy level of the virus. The virus will lose a set amount of energy with every run, and when it runs out of energy, it dies (i.e. it disappears). In order to survive, a virus needs to pick up energy, which it can only do by degrading the image. The more it changes the color of a pixel, the more energy it acquires. The difference between the color before and after is calculated.

We can see from a virus's behavior and direction whether it will be more or less adaptable - more or less able to survive.

There is a maximum number of viruses that can be present simultaneously (usually 1000). When that number is reached, the 'divide' instruction is ignored. If the virus has enough energy it will move around randomly, otherwise it will follow its favorite color and absorb part of the red component of the pixel it is on.

The Dynamics of the System

A viral attack will generally develop as follows:

a) A world is created from an image.

b) A population of viruses is generated randomly and introduced into the image. Every virus takes on its very own behavior, as the program defines them fortuitously.

c) Once the viruses have been placed in an image, the attack can start. It will consist of a series of action cycles that will only come to an end when there is no virus left alive (or after a given time limit).

Many different dynamics can be seen that depend on the parameterization of the experiment. For example:

a) Filtering: The viruses act as local filters on the image. The modification of a pixel's color will influence the subsequent dynamics, and it can influence the virus's demeanor and its trajectory. For example, a virus attracted by the intensity of the red component in pixels reduces the intensity of the color if it executes 'eat red' - and will tend to avoid the areas it has already visited. In order to have some control over the global effect, Joseph Nechvatal can define a set of active instructions; those which will be part of the genome program for the virus.

b) Reproduction, Evolution, and Adaptation: The instruction 'divide' will reproduce replicas of a virus (slightly different through mutation). The creation of these replicas will immediately trigger a considerable loss of energy in the virus. This means that a virus that is not capable of drawing energy from its environment will not survive much longer after it has carried out the 'divide' mandate. And so will its replicas.

On the other hand, the fact that these replicas are not identical to the original offers the possibility of examining new types of behavior.

When an adapted individual appears, it can remain in the image for quite some while. If it executes the 'divide' instruction, its descendants will most probably be equally adapted. The number of these agents will generally increase exponentially, and thereby create a large population of very active viruses.